Deep Structured Learning (IST, Fall 2018)

Homework 2

Instructor: André Martins
TAs: Vlad Niculae and Erick Fonseca

Deadline: Wednesday, October 31, 2018.

Please turn in the answers to the questions below together with the code you implemented to solve them (when applicable). Please email your solutions in **electronic format** (a single zip file) with the subject "Homework 2" to:

`deep-structured-learning-instructors@googlegroups.com`

Hard copies will not be accepted.

Question 1

OCR with an autodiff toolkit. In homework 1, you were asked to implemented a optical character recognition system from scratch. In particular, you had to wrote gradient backpropagation by hand. This time, you will implement the same system using a deep learning framework with automatic differentiation (suggested: Pytorch).

1. (10 points) Implement a linear model with ℓ_2-regularized logistic regression, using stochastic gradient descent as your training algorithm. Tune the following hyperparameters in your validation data:

    - The number of epochs.
    - The learning rate.
    - The regularization constant.

    For the best configuration, plot two things: the training loss and the validation accuracy, both as a function of the epoch number. Report the final accuracy in the test set.

2. (15 points) Implement a feed-forward neural network with a single layer, using dropout regularization. Try several activation functions (sigmoid, `tanh`, and `relu`), and different optimizers (SGD, Adam, Adagrad). For each choice, tune all the hyperparameters in the validation set (number of epochs, learning rate, number of hidden units, dropout probability). Make similar plots as in the previous question, and report the final test accuracy.

3. (5 points) Pick your favourite activation and optimizer and increase the number of layers. Make similar plots as in the previous question, and report the final test accuracy.

	Sunny	Windy	Rainy
Surf	0.4	0.5	0.1
Beach	0.4	0.1	0.1
Videogame	0.1	0.2	0.3
Study	0.1	0.2	0.5

Tabela 1: Emission probabilities: rows conditioned on columns.

	Sunny	Windy	Rainy
Sunny	0.7	0.3	0.2
Windy	0.2	0.5	0.3
Rainy	0.1	0.2	0.5

Tabela 2: Transition probabilities: rows (timestep $t + 1$) conditioned on columns (timestep t).

Question 2

Dynamic Programming. Your friend "John the Dynamic" lives in Lisbon and, depending on the weather conditions, he enjoys surfing, going to the beach, playing videogames, and studying machine learning. His activities are governed by a hidden Markov model, where the hidden variables correspond to the weather (Sunny, Windy, and Rainy) and the observed variables correspond to Surf, Beach, Videogame, and Study. The emission and transition probabilities are shown in Tables 1–2.

1. John's activities for the past week were like shown in Table 3. **Assume that the weather on October 7 was rainy, and on October 15 it was sunny.** Implement the Viterbi and forward-backward algorithms to answer the questions below.
   (a) (10 points) Knowing John's activities, what was the most likely weather for the past week?
   (b) (15 points) You made a bet with John where you receive 1€ for every day you guess the weather correctly, and you lose 1€ if your prediction is wrong. What would your bet (a) before knowing John's activities, but knowing that it was rainy on October 7 and (b) after observing John's activities and the weather in October 7 and October 15? In either case, what's your expected profit? Does more information make you richer?
2. (5 points) Actually, John never surfs two days in a row because (despite his nickname) he gets exhausted and he needs to rest at least one day before going back to the water. Can we accommodate this extra piece of knowledge in a hidden Markov model? Justify.

Monday, Oct 8	Videogame
Tuesday, Oct 9	Study
Wednesday, Oct 10	Study
Thursday, Oct 11	Surf
Friday, Oct 12	Beach
Saturday, Oct 13	Videogame
Sunday, Oct 14	Beach

Tabela 3: John's activities for the past week.

Question 3

Sequential OCR. So far, all your OCR experiments used models that try to predict each character independently from the others. In this exercise, you will solve the problem with structured prediction, using a linear sequential model.

1. (25 points) Take your perceptron implementation for homework 1 and change it to become a structured perceptron, exploiting the sequential structure of the characters (as they form words). As unigram features, use the pairwise features for pixels you used in homework 1. As bigram features, use only the conjunction of the two consecutive labels with no dependency on the pixels (i.e., a total of 26^2 bigram features). How does test accuracy compares with not using any structure? Hint: use your Viterbi implementation from the previous exercise, and don't forget to account for the start and stop symbols.

2. (15 points) Repeat the exercise above using a conditional random field (trained with stochastic gradient descent) instead of the structured perceptron.